

GITOPS AVEC KUBERNETES (K8S), TERRAFORM, AZURE ET AWS

Durée : 1 jour (7 heures)

CONNAISSANCES PREALABLES

- Connaissances générales des systèmes Linux.
- Notions de Git et gestion de versions.
- Compréhension des concepts DevOps.
- Connaissances de base de Kubernetes et du Cloud.
- Une première expérience avec Terraform constitue un avantage.

PROFIL DES STAGIAIRES

- Ingénieurs DevOps.
- Administrateurs Cloud.
- Administrateurs Kubernetes.
- Architectes Cloud.
- Ingénieurs plateformes.
- Responsables infrastructures.
- Développeurs impliqués dans les déploiements.
- Toute personne souhaitant découvrir les pratiques GitOps.

OBJECTIFS

À l'issue de cette formation, les participants seront capables de :

- Comprendre les principes fondamentaux du GitOps.
- Identifier les différences entre DevOps traditionnel et GitOps.
- Utiliser Git comme source de vérité des infrastructures.
- Comprendre l'intégration entre GitOps, Kubernetes et Terraform.
- Découvrir les mécanismes de synchronisation automatique.
- Déployer des applications selon une approche GitOps.
- Comprendre l'automatisation des infrastructures Cloud AWS et Azure.
- Identifier les bénéfices en matière de gouvernance, sécurité et traçabilité.
- Mettre en œuvre les bonnes pratiques GitOps dans un environnement Cloud Native.

CERTIFICATION PREPAREE

Aucune

METHODES PEDAGOGIQUES

- Présentations interactives.
- Démonstrations techniques.
- Travaux pratiques guidés.

- Études de cas.
- Ateliers d'automatisation.
- Retours d'expérience terrain.
- Support pédagogique remis aux participants.

FORMATEUR

- Consultant expert DevOps, Kubernetes, Cloud et Infrastructure as Code, disposant d'une expérience significative dans la mise en œuvre de plateformes GitOps en environnement AWS, Azure et Kubernetes.

METHODE D'EVALUATION DES ACQUIS

- Quiz de positionnement.
- Exercices pratiques.
- Études de cas.
- Ateliers GitOps.
- Évaluation finale des acquis.
- Débriefing collectif.

CONTENU DU COURS

Module 1 : Comprendre GitOps et son intégration dans l'écosystème Cloud Native (2h)

Objectifs

- Comprendre les fondements du GitOps.
- Identifier les composants d'une architecture GitOps moderne.

Contenu

Introduction au GitOps

- Origine et principes.
- Git comme source unique de vérité.
- Gestion déclarative des infrastructures.

GitOps et DevOps

- Complémentarités.
- Différences d'approche.
- Gouvernance des changements.

Les composants d'une architecture GitOps

- Dépôts Git.
- Pipelines CI/CD.
- Agents de synchronisation.
- Clusters Kubernetes.

Les bénéfices du GitOps

- Traçabilité.
- Auditabilité.
- Réversibilité.
- Sécurité.

Mise en pratique

Atelier :

Analyse d'un processus de déploiement traditionnel et identification des bénéfices d'une approche GitOps.

Module 2 : GitOps avec Kubernetes et Terraform (2h30)

Objectifs

- Comprendre l'automatisation des infrastructures et des applications.
- Découvrir les outils GitOps couramment utilisés.

Contenu

GitOps et Kubernetes

- Gestion déclarative des ressources.
- Synchronisation automatique des clusters.
- Gestion des déploiements applicatifs.

Intégration avec Terraform

- Infrastructure as Code.
- Gestion des ressources Cloud.
- Complémentarité GitOps et Terraform.

Présentation des principaux outils GitOps

- Argo CD
- Flux
- Terraform.
- Git.

Cycle de vie d'un changement

- Commit.
- Validation.
- Synchronisation.
- Déploiement.

Mise en pratique

Travaux pratiques :

Simulation d'un workflow GitOps intégrant Kubernetes et Terraform.

Module 3 : Automatiser les déploiements AWS et Azure avec GitOps (2h30)

Objectifs

- Comprendre l'intégration du GitOps dans les environnements Cloud.
- Mettre en œuvre les bonnes pratiques de gouvernance et d'automatisation.

Contenu

GitOps dans AWS

- Gestion des infrastructures Cloud.
- Déploiement automatisé des ressources.
- Gouvernance et contrôle des changements.

GitOps dans Azure

- Gestion des ressources Azure.
- Automatisation des déploiements.
- Intégration avec les services Cloud Native.

Sécurité et gouvernance

- Gestion des accès.
- Gestion des secrets.
- Audit des changements.

Bonnes pratiques GitOps

- Organisation des dépôts.
- Gestion des environnements.
- Validation des changements.
- Industrialisation.

Mise en pratique

Cas fil rouge :

Conception d'une chaîne GitOps complète intégrant :

- Dépôt Git centralisé.
- Gestion des ressources Terraform.
- Déploiement Kubernetes.
- Synchronisation automatisée.
- Gouvernance des changements.
- Déploiement multi-environnements AWS et Azure.

Notre référent handicap se tient à votre disposition au [01.71.19.70.30](tel:01.71.19.70.30) ou par mail à <mailto:referent.handicap@edugroupe.com> pour recueillir vos éventuels besoins d'aménagements, afin de vous offrir la meilleure expérience possible.